ISSN:0975-9646

Multi labelled Graph Mining with Dynamic Network

Ankita V. Raiyani
Alpha College of Engineering and Technology,
Gandhinagar,
Gujarat

Prof. Ajaykumar T. Shah
Alpha College of Engineering and Technology,
Gandhinagar,
Gujarat

***Abstract*-Graph represents the real world problem with better clarity and more informative. Graph mining has various sub domains including matching which is more prominent algorithm. In this paper, we propose the multi labelled graph matching with dynamic network. Multi labelled graph give more information about that graph and also dynamic graph represent that real time applications. We apply our approach on social network to understand how the social network changes over time. Experimental evaluation of that algorithm in comparison with some recent algorithms in the field highlights that superior performance.**

I.INTRODUCTION

Graph mining and network analytics is critical to a variation of application domains, fluctuating from community detection in social networks, malicious program analysis in computer security, to examine for functional modules in biological pathways and structural analysis in chemical complexes. There is an emerging need to systematically investigate the modelling, managing, and mining of large-scale graphs and networks in bioinformatics, social networks, and computer systems. Given a graph G, a matching M is a set of edges such that no two edges in M are incident on the same vertex. Matching is fundamental combinatorial problem that has applications in many contexts: high-performance computing, bioinformatics, network switch design, web technologies, etc. Examples in the first context include sparse linear systems, where matching are used to place large matrix elements on or close to the diagonal, block triangular decomposition, computing a sparse basis for the null space or column space of under-determined matrices, and multi-level graph partitioning algorithms where matching are used in the coarsening phase. Graph matching is most important element in graph based pattern recognition application. The similarity between two real vectors are easily defined but it is not easy to define how similar between two graphs. The challenge was to be able to import the whole set of learning and recognition tool in domains of graph. This can be achieved by defining a measure of dissimilarity directly in graph domains and through a representation of them in suitable space. There are classified in two well defined in exact or inexact matching. The Boolean evaluation of similarity of graph and more complex problem is how much they differ.The matching problems can befurther classified intobased on the objective function:

1. Maximum (Cardinality) Matching: Maximize the number of edges in the matching.
2. Maximum (Edge) Weighted Matching: Maximize the sum of the weights of the matched edges.
3. Maximum (Vertex) Weighted Matching: Maximize the sum of the weights of the matched vertices.

Matching in a bipartite graph is easier to compute than in general (or no bipartite) graphs. Similarly, the unweighted versions are easier than the weighted versions of the matching problem. The weighted versions may also have additional restrictions on the cardinality of the matching, e.g., a maximum weight matching among all matching of maximum cardinality.A complete matching, every vertex of the graph is incident to exactly one edge of the matching. A perfect matching is therefore a matching of a graph containing n/2 edges; the largest possible, meaning perfect matching's only possible on graphs with an even number of vertices. A perfect matching is sometimes called a complete matching or 1-factor.The nine perfect matching's of the cubical graph is illustrated below.

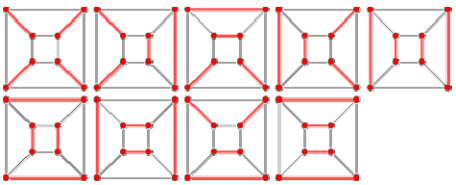

Fig 1. Perfect Matching of Cubic Graph

The class of graphs known as perfect graphs is distinct from the class of graphs with perfect matching.

Dynamic graph containing the sequence of graph that changes over time that changes their structural as well as dynamic properties of network. We represent the networks including entities and relationship between the entities. For example, biological cells include various molecules and relationship between molecules. The biological cells are also active systems, which continuously change over the time. For analysis of the dynamic graph how that graph transformed to other graph also measure the difference between the two graphs. So our goal is also to find how and how much that graph is differ from each other.

II.GRAPH PRELIMINARY DEFINATIONS

In this section, we will give some preliminary definition, mainly regarding labelled graph, how to represent graph in different ways and so on.

A Graph is a set of Vertices and a set of Edges. G = (V, E). A vertex of a Graph is a connection point. A Graph has a set of Vertices, usually shown as V = {v_1, v_2,

...,v_n}where V = {A, B, C} or V = {1, 2, ... N}. The number of Vertices in a graph is |V|, but is sometimes written in equations as just V.A Vertex may have no connections, one connection or many connections. A vertex may have any number of properties such as a name or a colour.

An Edge in a graph is a connection between vertices. Given vertices v1 and v2 in a Graph, the edge between them may be written as (v_1,v_2) or sometimes [v_1,v_2].A Graph has a set of Edges usually denoted E. E = {(v_1, v_2), (v_2,v_3)}

The number of Edges in a graph is |E|, but is sometimes written in equations as just E.

A Graph is called Undirected if the edges have no implied direction. (v_1,v_2) is the same as (v_2,v_1), the edge just connects v_1 to v_2.

A Graph is called Directed if the edges have a direction. (v_1,v_2) means an edge starting at v_1 and going to v_2. (v_1,v_2) is NOT the same as (v_2,v_1).A Weighted Graph has edges with an additional property, a weight. Weights may be integers, real numbers, or any type of quantity. (v_1, v2, 10GPM) would indicate an Edge from vertex v_1 to v_2 with a flow of 10 gallons per minute. There are special cases that may not allow zero or negative weights.An Edge colour Graph has edges with an additional property, a color. In general, an Edge can have many properties that depend on what the graph represents. (v_1, v_2,10GPM, green).

A general graph may have a self loop which is an edge that goes from a vertex to itself, e.g. (v_3,v_3).A multi-graph may have more than one edge between the same pair of vertices, e.g. E = {(v_1,v_2), (v_1,v_2), (v_2,v_3)}.In a Directed Graph (v_1,v_2) and (v_2,v_1) is not a multi-graph, just two edges going different directions.

In an Undirected Graph sometimes (v_1, v_2) and (v_2,v_1) maybe included even though they are redundant and the Graph may not be considered a multi-graph.

Directed - the edges have a direction, usually drawn with an arrow head at one end. A DAG Directed Acyclic Graph is a restricted case with no cycles (no loops following direction of edges.)Undirected - usually the default if nothing else said. The edges do not have direction.Acyclic - no cycles, there is a Path that includes at least one edge that can return to the starting vertex.

Graph can be further classified into labelled or unlabelled. Labelled graph can be classified as vertex labelled, edge labelled and both vertex and edge labelled. Graph is multi labelled if its vertices or edges have more than one label. For Example when two people communicate with telephone, vertex represents the person name, id address etc. While edge represents the relationship between the person information like frequency of call, call duration etc.

III. PROBLEM STATEMENT AND PROPOSED METHOD

Graph matching plays important role in molecular application to social and communication network. Graph matching is the process of finding the occurrences of given pattern or given graph in large graph or structure. The given pattern graph, which is very smaller than large graph, is known as query graph, while large graph is known as reference graph. Labelled graph enhance the graph matching first that match the vertex and then match that label on vertex.

Graph matching in single label is simple and easy way but in multi labelled with dynamic network is not easy to define with exact matching.

Dynamic Multi labelled graph matching is enhance the multi labelled graph matching because that are used in real time application like social and computational chemistry that can also increase the complexity.

Graph matching performance depends upon the labelling, indexing, nature of inputs and matching process. Labelling means label only on vertex or edge or both. Some algorithm are designed to work on only vertex labelled graph while some are designed to work on only edge labelled graph or both. Here we design algorithm which is multi labelled graph matching with dynamic dataset.

The input graph has two attribute like size and type. That graph varies depend upon that application. For example, social network consist of thousands of vertices. So the type also play major role in algorithm. A single large graph or graph database with large numbers of small graph.

The indexing process in graph matching creates the indexing on their reference graph vertices for matching purpose. The different data structure is use in indexing method like tree based, path based or graph based. A path based structure is not suitable for large network because that cannot handle complex network since that information can be lost. The matching process involved in vertex matching as well edge matching and both.

This graph matching in multi labelled can be used in computational biology and social network analysis but that can handle dynamic network. That biology and social network are real time applications which are use dynamic nature of input.

This paper proposes the following algorithm.

1. Initialize the Reference graph.
2. Initialize the Query graph.
3. Generate the pattern list of Query graph using search strategy with Root vertex.
   Root vertex=highest degree of query graph.
4. Create the Match list of Reference graph.
5. Compare the Match list and pattern list with neighbourhood information and create number of matches.
6. Check with threshold.
   a. If above the threshold
      i. Create Add list of new Reference graph
      ii. Create the Remove List of new Reference Graph
      iii. Repeat step 4 and 5.
   b. Else
   i. Matches found.

On initialize the neighbourhood information of vertex. To initialize the query graph, this is smaller than the reference graph. We also generate the pattern list of the query graph using BFS search strategy. Here we use root

vertex which has highest degree of the query graph. Highest degree of the vertex can be calculated by neighbourhood information of query graph. In next step, we generate the match list, we compare reference graph vertex to query graph vertex and generate the match list of reference graph. Here we only compare the vertex matching, but in edge matching us compare the pattern list and match list with neighbourhoodinformationand generate the number of matches. Now we check for threshold, which is differ and depend on the database. If value is above threshold we generate the new Add list,here we compare the vertex with neighbourhood information of query graph to reference graph.Add list contains new reference graph and old reference graph common vertex and remaining vertex of new reference graph. Also we create Remove list of new reference graph. Remove list contains the vertex list which is removed from new reference graph. If any of the vertex which is part of match list and is also part of remove list we remove from match list.

Dynamic graph means graph that change their properties at run time. So we find that we find how one graph is differing from others. How that change that properties at runtime. We can check that dynamically properties in many ways like time manner, threshold value etc. In time manner we can check that graph in constant time and also that require lots of time for computation. So in this algorithm we use the threshold value which is depend upon the database. We generate the graph on biological network where vertices represent the compounds, genes, relations and reactions and edges represent that relationship between the vertices. That chemical compounds or genes change over time. Only when that amount of chemical compound that over the threshold value that shown in new reference graph.

IV. Implementation And Experimetal Evulation

We consider the MuGRAM, the latest algorithm in the graph matching for comparison of the proposed algorithm .These algorithm is evaluated using two data sets,DBLP Dataset and Enron Email Data set. The DBLP dataset, where vertex represents the author information like that id, name etc. while edge represents the publication information, conference information etc. Here we take threshold value is maximally author relationship information. We take another dataset Enron email dataset where vertex represent the employee information like emp id, name etc., while edge represent the email sent or receive by email accounts.

All experiments are carried out on windows 7system with 4 GB memory and Intel Core processor. We represents that table for Enron data set.

SR No	Query Size	Proposed Algorithm T(Sec)	MuGRAM T(Sec)
1	5	0.0373	0.280
2	10	0.0578	1.10
3	15	0.1564	1.820
4	20	0.2629	7.04

TABLE I EXPERIMENAL RESULTS WITH ENRON DATA SET

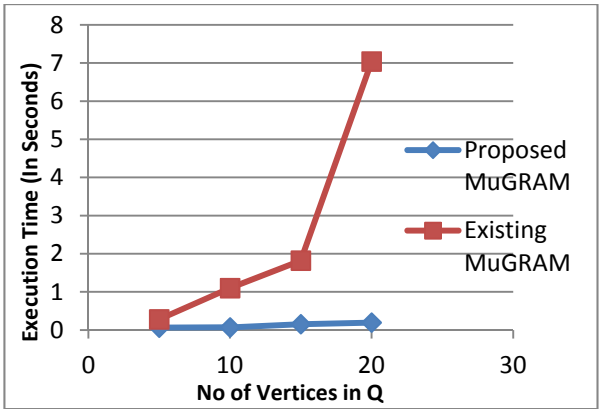

Fig 2. Execution time taken by MuGRAM and Proposed algorithm for graph matching

In the subsequent experimentation, the performance of the proposed algorithm on Dataset characterised by multi labelled graph was evaluated. Since multi label with dynamic network enhance the performance of the multi labelled graph with vertex matching as well as edge matching with in exact matching.

V. Conclusion And Future Work

Graph matching domain addresses the real time application like the computational biology and social network etc. This application is not for static network but it is used for dynamic network properties.The proposed algorithm is evaluated along with MuGRAM with same datasets. The proposed Algorithm found faster as compare to all algorithm. There is also further enhancement of that algorithm by paralizing the matching process. This algorithm scope can cover the social network and communication network.

References

[1] HakanKardes , Mehmet HadiGunes: Structural Graph Indexing for Mining Complex Networks. InIEEE 30thInternational Conference on Distributed Computing Systems (2010) 99-104

[2] Yuhua Li, Quan Lin, Gang Zhong, Dongsheng Duan,Yanan Jin: A Directed Labeled Graph Frequent Pattern Mining Algorithm Based on Minimum Code. In: 3rd International Conference on Multimediaand Ubiquitous Engineering (2009) 353-359

[3] Winnei W.M. Lam,Keith C.C. Chan, Analysing Web Layout Structures using Graph Mining.In : IEEE international Conference on data mining(2009)

[4] Swapnil Shrivastava and SupriyaN.Pal: Graph Mining Framework for Finding and Visualizing Substructures Using Graph Database. In: Advance in Social Network Analysis and Mining (2009) 379-380

[5] Chang Hau You, Lawerence Holder, Diane Cook, Graph Based Data Mining in Dynamic Networks: ComparisonEmpirical of Compression based on based on frequency sub graph Mining .In: IEEE International Conference on data mining (2008) 929-938

[6] Winnie W M Lam, Keith C C Chan, Analyzing Web Layout Structure using Graph Mining. In: IEEE International Conference on data mining (2010)

[7] Guanling Lee, Sheng Lung Peng, Shih Wei Kuo and Yi- Chun Chen, Mining Frequent Maximal Cliques Efficiently by Global View Graph . In: IEEE (2012) 1362-1366

[8] Seema Desai, Satish R Devane, VimlaJethani, Association Rule Mining Using Graph and clustering Technique. In: IEEE (2012) 893-897

[9] Saif Ur Rehman, AsmatUllah Khan, Simon Fong, Graph Mining: A Survey of Graph Mining Techniques. In IEEE(2012) 88-92

[10] ZhaonianZou, Jianzhong Li, Member, IEEE, Hong Gao, and ShuoZhang : Mining Frequent Subgraph Patternsfrom Uncertain Graph Data. InIEEE Transactions on knowledge and data Engineering, (2010)1203-1218
[11] *http://www.rcsb.org/pdb/home/home.do*
[12] W. Cohen, "Enron email Dataset (2005),"URL *http://www.cs.cmu.edu/enron*
[13] Lorenzo livi, AntonelloRizzi: The Graph Matching problem. In Springer (2012)253-283
[14] *http://www.jhowell.net/cf/scores/scoreindex.html*